



INSTITUTO POLITÉCNICO NACIONAL

ESCUELA SUPERIOR DE CÓMPUTO

NOMBRES:

PADILLA MATIAS CRISTIAN MICHEL

SAUCILLO GONZÁLEZ JESSE OBED

GRUPO:

4BM1

TRABAJO:

Practica 5

"Razonamiento basado en Reglas"

MATERIA:

Fundamentos de Inteligencia Artificial

FECHA

31 de octubre del 2023

INSTITUTO POLITÉCNICO NACIONAL



ESCOM

PRACTICA 5. Razonamiento basado en reglas

RESUMEN

En esta practica, se desarrollara un sistema experto botánico basado en razonamiento lógico y reglas, implementado en el lenguaje Prolog. El sistema debera tener la capacidad de identificar diversas plantas a partir de las características proporcionadas por el usuario. Utilizando un conjunto de reglas definidas, el sistema solicita información específica sobre la planta y, mediante el razonamiento basado en reglas, infiere la identidad de la planta con precisión. Este proyecto demuestra la eficacia del razonamiento basado en reglas para la construcción de sistemas expertos en el campo de la botánica. El trabajo refleja la capacidad del sistema para realizar diagnósticos precisos y sienta las bases para futuras mejoras y expansiones en la base de conocimientos.

INTRODUCCIÓN

En el campo de la inteligencia artificial, el razonamiento basado en reglas es una técnica fundamental para la construcción de sistemas expertos. Los sistemas expertos son programas de computadora que emulan la habilidad de un experto humano en un dominio específico, utilizando reglas y lógica para tomar decisiones y ofrecer soluciones a problemas complejos. En esta práctica, exploramos el poder del razonamiento basado en reglas a través de la implementación de un sistema experto para un consultorio botánico.

DESARROLLO

El código proporcionado implementa un sistema experto para un consultorio botánico en Prolog, un lenguaje de programación basado en lógica. A continuación, se describe parte del código:

Inicio del Programa:

El programa comienza con la definición de hechos y reglas dinámicas utilizando `dynamic conocido/1`. Esto permite la manipulación dinámica de la base de conocimientos durante la ejecución del programa.

```
:- dynamic conocido/1.
```

Menú Principal:

Se presenta un menú al usuario, proporcionando la opción de realizar una consulta botánica o salir del programa. Dependiendo de la elección del usuario, el programa procede con la consulta o finaliza la ejecución.

```
main :- bienvenida,  
      nl, write('Menu'), nl,  
      nl, write('1 Consulta botánica'),  
      nl, write('2 Salir'),  
      nl, nl, write('Indica tu opción: '),  
      read(Opcion),  
      nl,  
      (Opcion = 1 -> consulta ; Opcion = 2, writeln('Gracias por no dejarme plan  
tado, ten un plantabuloso día ^.\n')).
```

Búsqueda y Diagnóstico:

En la consulta, el sistema hace preguntas al usuario sobre las características de la planta en cuestión. El predicado `haz_diagnostico/1` busca una planta que coincida con las características proporcionadas. Utiliza las reglas definidas en la base de conocimientos para obtener hipótesis y síntomas asociados a la planta.

```

consulta :- limpia_memoria_de_trabajo,
              writeln("\nResponderé algunas preguntas. Por favor, responde con 'si'
o 'no'\n"),
              haz_diagnostico(X),
              escribe_diagnostico(X),
              ofrece_explicacion_diagnostico(X).

consulta :- write('No hay suficiente información para hacer una identificación.').

% Realiza el diagnóstico basado en los síntomas ingresados
haz_diagnostico(Planta) :- obten_hipotesis_y_sintomas(Planta, ListaDeCaracteristic
as),
                          prueba_presencia_de(Planta, ListaDeCaracteristicas).

```

Pruebas de Presencia:

La presencia de los síntomas se verifica utilizando el predicado `prueba_presencia_de/2`, que busca en la memoria de trabajo (conocido/1) si los síntomas ya han sido confirmados.

```

prueba_presencia_de(_, []).

prueba_presencia_de(Planta, [Caracteristica | RestoCaracteristicas]) :-
    prueba_verdad_de(Planta, Caracteristica),
    prueba_presencia_de(Planta, RestoCaracteristicas).

```

Preguntas y Proceso de Respuestas:

Las preguntas sobre las características se realizan al usuario utilizando el predicado `pregunta_sobre/3`. El programa procesa las respuestas (si, no o porque) utilizando el predicado `process/4`, el cual maneja la lógica del razonamiento basado en reglas.

```

pregunta_sobre(Planta, Caracteristica, Respuesta) :-
    write('¿La planta tiene '), write(Caracteristica), write('? '),
    read(Respuesta),
    process(Planta, Caracteristica, Respuesta, Respuesta).

```

Explicación del Diagnóstico:

Después de hacer el diagnóstico, el programa ofrece una explicación al usuario si este lo solicita, detallando las características que llevaron a la identificación de la planta.

```

ofrece_explicacion_diagnostico(Planta) :-
    pregunta_si_necesita_explicacion(Respuesta),
    actua_consecuentemente(Planta, Respuesta).

pregunta_si_necesita_explicacion(Respuesta) :-
    write('\n¿Quieres saber por qué llegué a esta conclusión? '),
    read(RespuestaUsuario),
    asegura_respuesta_si_o_no(RespuestaUsuario, Respuesta).

```

Memoria de Trabajo:

La memoria de trabajo (conocido/1) se utiliza para almacenar los síntomas confirmados durante la consulta. Se limpia al inicio de cada consulta para evitar datos obsoletos.

```

limpia_memoria_de_trabajo :- retractall(conocido(_)).

```

Base de Conocimientos:

Las reglas de diagnóstico se definen en la base de conocimientos, donde cada planta tiene asociadas ciertas características. Se han proporcionado ejemplos de plantas como 'Rosa', 'Orquídea', 'Cactus', etc., junto con sus características distintivas.

```

reglas(planta('Rosa'), características(['tiene espinas', 'tiene flores de colores'])).
reglas(planta('Orquídea'), características(['tiene flores exóticas', 'crece en climas húmedos'])).
reglas(planta('Cactus'), características(['almacena agua en sus hojas', 'tiene espinas gruesas'])).
reglas(planta('Pino'), características(['tiene hojas en forma de aguja', 'produce piñas'])).
reglas(planta('Cocotero'), características(['da cocos', 'tiene hojas largas y flexibles'])).
reglas(planta('Lirio'), características(['tiene flores grandes y vistosas', 'crece en zonas húmedas'])).
reglas(planta('Helecho'), características(['tiene hojas verdes y frondosas', 'prefiere sombra y humedad'])).
reglas(planta('Tulipán'), características(['tiene flores en forma de copa', 'viene en varios colores', 'es muy apreciada en jardinería'])).
reglas(planta('Suculenta'), características(['almacena agua en sus hojas o tallos', 'tolera condiciones áridas', 'fácil de cuidar'])).
reglas(planta('Bambú'), características(['crece rápidamente', 'tiene tallos huecos', 'se utiliza en construcciones y mobiliario'])).
reglas(planta('Girasol'), características(['tiene flores grandes y amarillas', 'sigue la dirección del sol', 'se utiliza para producir aceite comestible'])).

```

El programa solicita al usuario respuestas a preguntas específicas sobre las características de la planta y utiliza el razonamiento basado en reglas para inferir y proporcionar una identificación precisa.

DEMOSTRACIÓN DE FUNCIONAMIENTO

Ejecutando el programa

Pantalla de inicio:

```
Bienvenido a mi Consultorio Botánico "El plantastico" :D
=====

Puedo ayudarte a identificar plantas, pero ten en cuenta
que puedo regarla, solo soy un fanatico de las plantas :3

Menu

1 Consulta botánica
2 Salir

Indica tu opción: 1
```

Al escoger la opción 1:

```
Responderé algunas preguntas. Por favor, responde con 'si' o 'no'

¿La planta tiene tiene espinas? |: si.
¿La planta tiene tiene flores de colores? |: no.
¿La planta tiene tiene flores exóticas? |: si.
¿La planta tiene crece en climas húmedos? |: no.
¿La planta tiene almacena agua en sus hojas? |: si.
¿La planta tiene tiene espinas gruesas? |: si.

La planta identificada es Cactus.

¿Quieres saber por qué llegué a esta conclusión? |:
```

Al responder positivo a la explicación:

```
¿Quieres saber por qué llegué a esta conclusión? |: si.

La identificación se basa en las siguientes características:

almacena agua en sus hojas
tiene espinas gruesas
```

Al escoger la opción 2:

Bienvenido a mi Consultorio Botánico "El plantastico" :D
=====

Puedo ayudarte a identificar plantas, pero ten en cuenta
que puedo regarla, solo soy un fanatico de las plantas :3

Menu

1 Consulta botánica
2 Salir

Indica tu opción: 2
|: .

Gracias por no dejarme plantado, ten un plantabuloso dia ^.^

CONCLUSIONES

El razonamiento basado en reglas ha demostrado ser una herramienta poderosa para la construcción de sistemas expertos en el ámbito de la botánica. El sistema experto desarrollado en esta práctica puede identificar diversas plantas basándose en las características proporcionadas, ofreciendo una muestra clara de cómo las reglas lógicas pueden emplearse para simular el proceso de toma de decisiones humano.

BIBLIOGRAFÍA

Bratko, I. (2012). Prolog Programming for Artificial Intelligence. Pearson Education.

Russel, S., & Norvig, P. (2016). Artificial Intelligence: A Modern Approach. Pearson.